

05BM - Fundamentos de Ingeniería del Software.

BLOQUE 1.- Introducción

TEMA 1.- Introducción a la ingeniería del software

Departamento de Informática y Sistemas

Facultad de Informática

Campus Universitario de Espinardo – Murcia

Profesor: Juan Antonio López Quesada





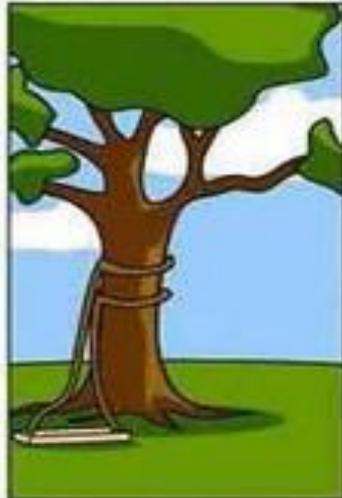
La solicitud del usuario



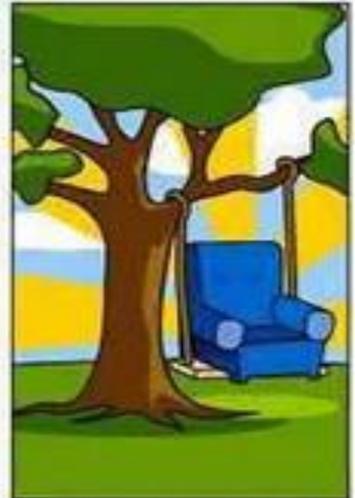
Lo que entendió el líder del proyecto



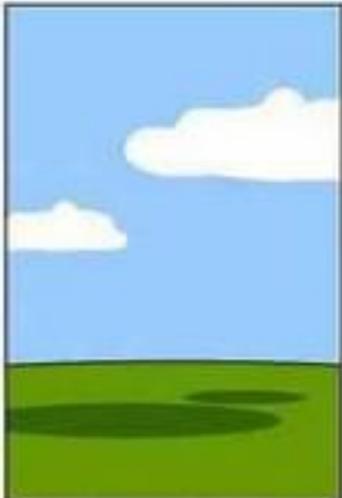
El diseño del analista de sistemas



El enfoque del programador



La recomendación del consultor externo



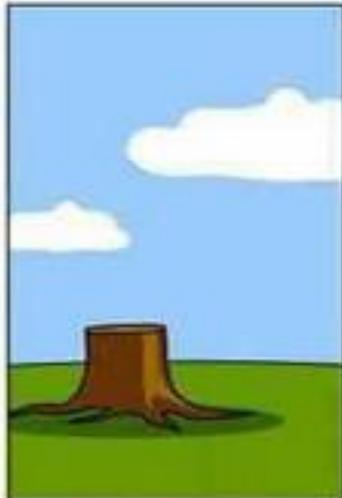
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

ÍNDICE DE CONTENIDOS



ÍNDICE DE CONTENIDOS

1.- Introducción a la Ingeniería del Software

1.1.- Un poco de historia....

1.2.- Factores de calidad del software.

1.3.- Problemas en el desarrollo de software.

1.4.- La Ingeniería del Software.

1.5.- Visión general del proceso de Ingeniería del Software.

2.- Sistema de Información.

2.1.- Concepto de sistema.

2.2.- Sistemas de información (SI).

2.3.- Sistemas de información automatizado y empresarial.

2.4.- Seguridad y protección de la Información.

3.- Herramientas CASE.

BIBLIOGRAFÍA

- ✦ (Pressman 2001) o (Pressman 1998) Cap.1 y Cap. 2 (aptdo. 2.1).
- ✦ (Piattini et al. 96) cap.1
- ✦ En el cap. 2 hay ejemplos de sistemas de información en la empresa.

1.- Introducción a la Ingeniería del Sw

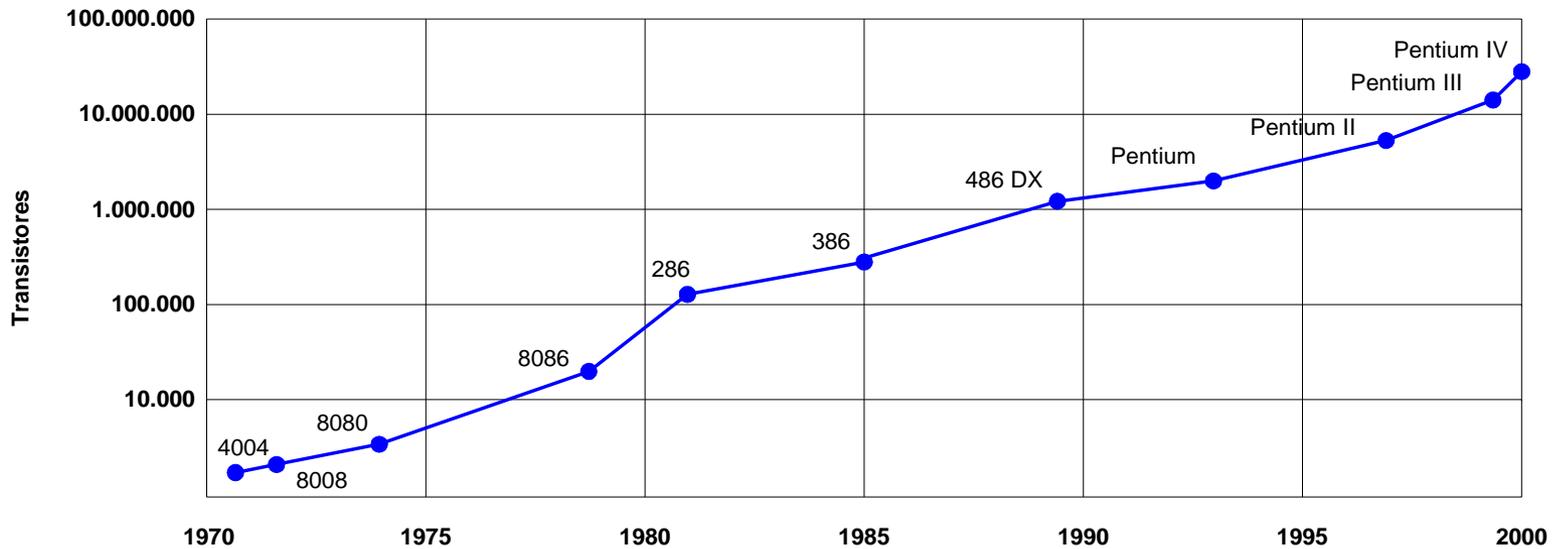
1.1.- Un poco de Historia.

- ❑ En menos de cuatro décadas nos hemos rodeado de máquinas capaces de procesar miles de millones de operaciones por segundo (MTOPS). En 1946 ENIAC ocupaba una superficie de 160 m², pesaba 30 toneladas, y ofrecía una capacidad de proceso de 30.000 instrucciones por segundo. En 2002 El microprocesador Pentium IV a 2 Ghz ocupa una superficie de 217 mm² y tiene una capacidad de proceso de 5.300 MTOPS (“Millions of theoretical operations per second”).
- ❑ El incremento constante de la capacidad de operación, miniaturización y reducción de costes para la producción de hardware; y a éstos se ha sumado en la última década el avance de las comunicaciones entre sistemas, a tenido como consecuencia la difusión de: ordenadores potentes, que pueden llevarse en el bolsillo y en permanente conexión con grandes sistemas, redes de comunicación públicas, sistemas de localización GPS, etc.

Este es el escenario creado por la industria del hardware, y que en las tres últimas décadas ha implicado a los desarrolladores de software en retos a los que, ¿han sabido responder con solvencia?.

1.- Introducción a la Ingeniería del Sw

1.1.- Un poco de Historia.

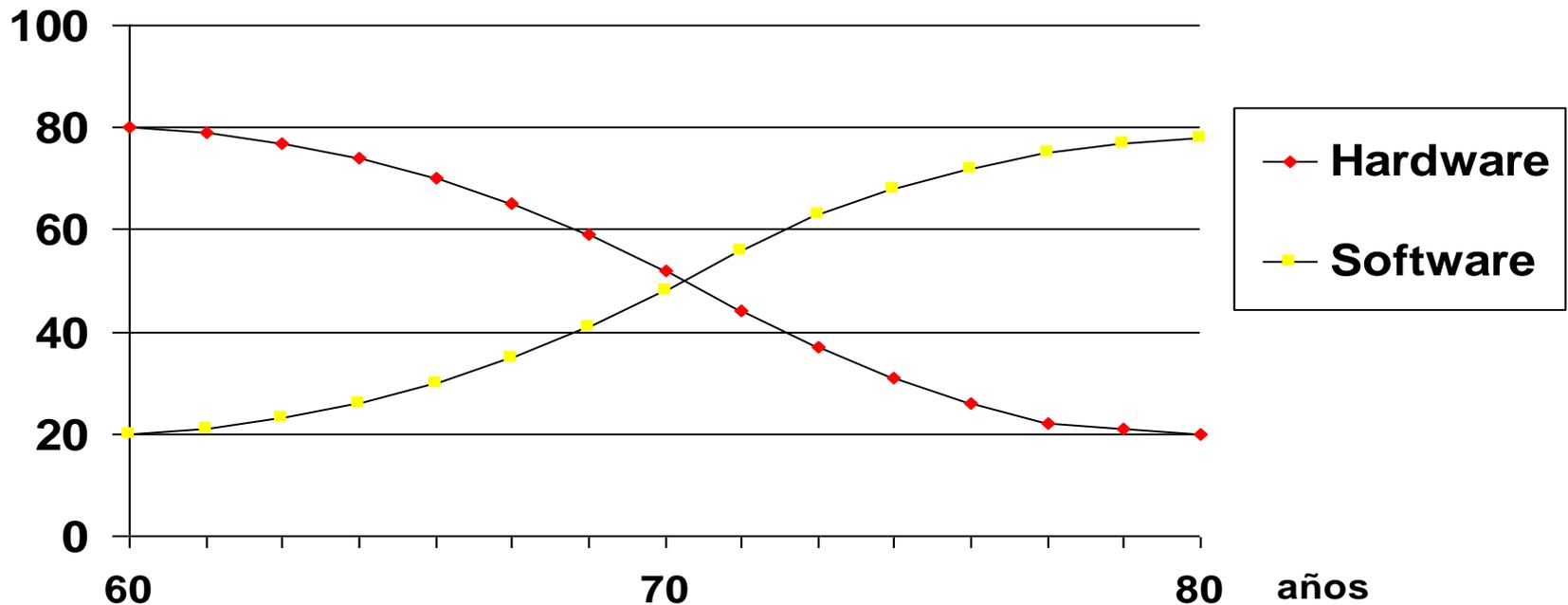


- Incremento de la capacidad de operación.
- Incremento de la miniaturización.
- Reducción de costes en la producción.

1.- Introducción a la Ingeniería del Sw

1.1.- Un poco de Historia.

Porcentaje del coste total del sistema



1.- Introducción a la Ingeniería del Sw

1.1.- Un poco de Historia.

Este problema se identificó por primera vez en 1968, año en el que la organización NATO desarrolló la primera conferencia sobre desarrollo de software, y en la que se acuñaron los términos **“crisis del software”** para definir a los problemas que surgían en el desarrollo de sistemas de software, e **“ingeniería del software”** para describir el conjunto de conocimientos que existían en aquel estado inicial.

Algunas referencias útiles para comprender cuáles eran los conocimientos estables para el desarrollo de software en 1968 son:

- En 1962 se publicó el primer algoritmo para búsquedas binarias.
- C. Böhm y G. Jacopini publicaron en 1966 el documento que creaba una fundación para la eliminación de “GoTo” y la creación de la programación estructurada.
- En 1968 los programadores se debatían entre el uso de la sentencia GoTo, y la nueva idea de programación estructurada; ese era el caldo de cultivo en el que Edsger Dijkstra escribió su famosa carta “GoTo Statement Considered Harmful” en 1968.
- La primera publicación sobre programación estructurada no vio la luz hasta 1974, publicada por Larry Constantine, Glenford Myers y Wayne Stevens.
- El primer libro sobre métrica de software fue publicado en 1977 por Tom Gilb.
- El primero sobre análisis de requisitos apareció en 1979.

1.- Introducción a la Ingeniería del Sw

1.1.- Un poco de Historia.

Desde 1968 hasta la fecha han sido muchos los esfuerzos realizados por los departamentos de informática de las universidades, y por organismos de estandarización (SEI, IEEE, ISO) para identificar las causas del problema y **definir pautas estándar para la producción y mantenimiento** del software.

Los esfuerzos se han encaminado en tres direcciones principales.

- Identificación de los factores clave que determinan la calidad del software.
- Identificación de los procesos necesarios para producir y mantener software.
- Acotación, estructuración y desarrollo de la **base de conocimiento** necesaria para la producción y mantenimiento de software.

El resultado ha sido la necesidad de profesionalizar el desarrollo, mantenimiento y operación de los sistemas de software, introduciendo métodos y formas de trabajo sistemáticos, disciplinados y cuantificables.

1.- Introducción a la Ingeniería del Sw

1.2.- Factores de la Calidad del SW.

Elemento lógico, no físico.

Desarrollado, no 'fabricado'.

No se 'estropea', ¡se deteriora!

(deterioro por 'cambios' - Mantenimiento)

Mayoritariamente cerrado:

(poco ensamblaje de componentes:

Herramienta de reutilización)

→ Metodología de Planificación,
Desarrollo y Mantenimiento de sistemas de información
MÉTRICA. Versión 3

Introducción

Planificación de Sistemas de Información (Proceso PSI)

Estudio de Viabilidad del Sistema (Proceso EVS)

Análisis del Sistema de Información (Proceso ASI)

Diseño del Sistema de Información (Proceso DSI)

Construcción del Sistema de Información (Proceso CSI)

Implantación y Aceptación del Sistema (Proceso IAS)

Mantenimiento del Sistema de Información (Proceso MSI)

- ❑ Fiable, Eficiente, Robusto, Correcto, Portable, Adaptable (extensibilidad), Inteligible, No Erróneo, Reutilizable (reusabilidad), Mantenibilidad, Confiabilidad, Eficiencia....

1.- Introducción a la Ingeniería del Sw

1.3.- Problemas en el desarrollo de software.

- ❑ Incapacidad para estimar tiempo, coste y esfuerzo para el desarrollo de un producto software.

**Metodología de Planificación,
Desarrollo y Mantenimiento de sistemas de información**
MÉTRICA. Versión 3

¿Solución?

Planificación de Sistemas de Información (Proceso PSI)
Estudio de Viabilidad del Sistema (Proceso EVS)
Aspectos de gestión mediante sus INTERFACES.

- ❑ Falta de calidad del producto software.

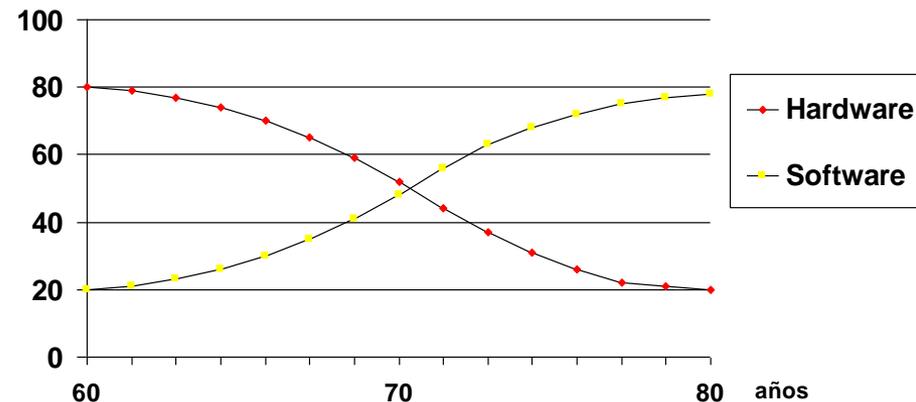
¿Solución?

**Metodología de Planificación,
Desarrollo y Mantenimiento de sistemas de
información**

- ❑ Avance del hardware y necesidad de aplicaciones más complejas.

⇒ Coste hardware/software.

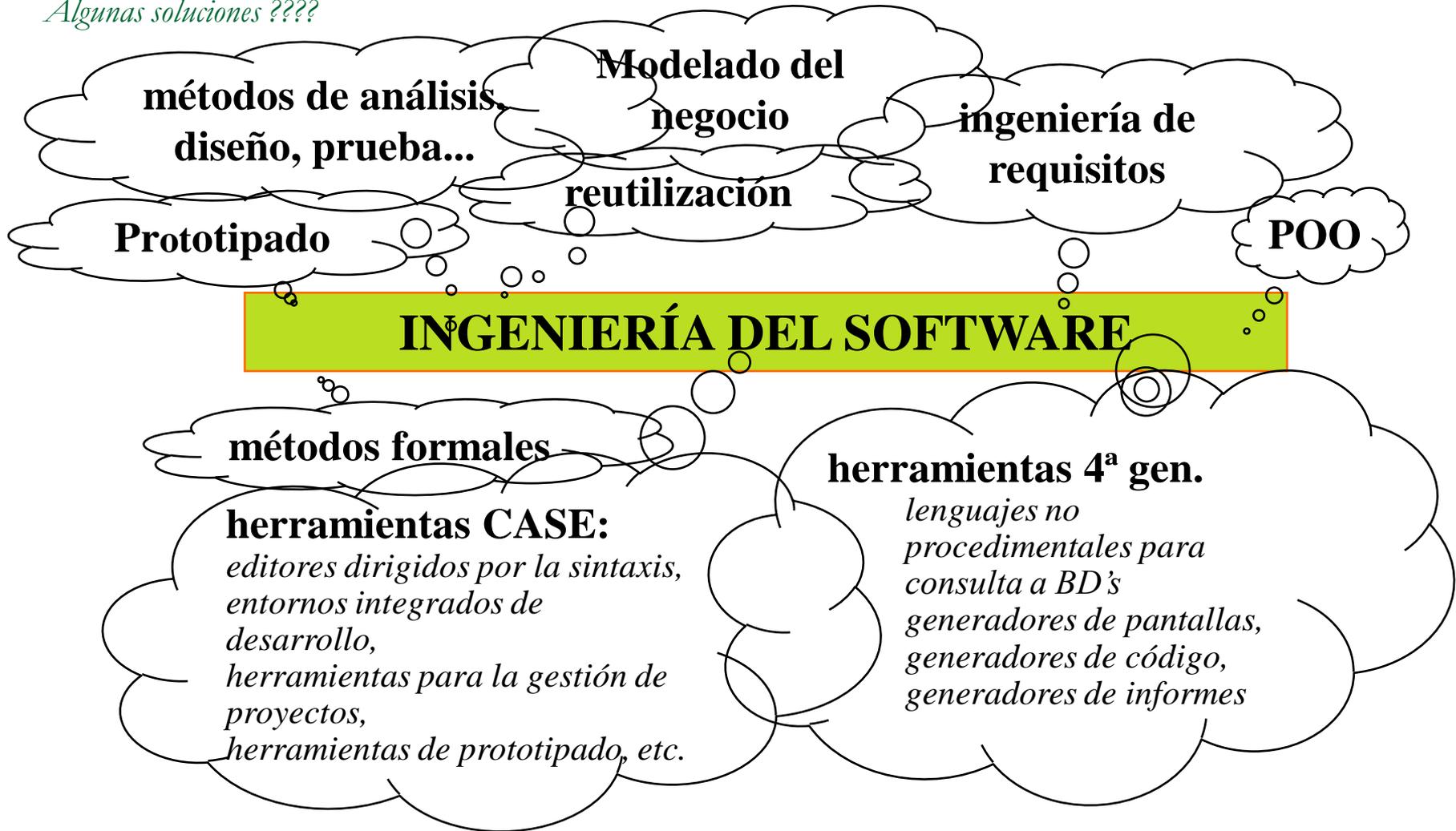
de
Porcentaje del coste
total del sistema



1.- Introducción a la Ingeniería del Sw

1.3.- Problemas en el desarrollo de software.

Algunas soluciones ????



1.- Introducción a la Ingeniería del Sw

1.4.- La Ingeniería del Sw.

Definirse a sí misma: ¿Cuáles son las áreas de conocimiento que la comprenden?

→ **SWEBOK: Software Engineering Body of knowledge:**

El cuerpo de conocimiento identificado por el proyecto SWEBOK se ha configurado como el estudio más relevante y como la referencia de más autoridad en toda la comunidad informática para la acotación y descripción de los conocimientos que configuran la Ingeniería del software.

Definir los procesos que intervienen en el desarrollo, mantenimiento y operación del software

→ **ISO/IEC 12207: Procesos del ciclo de vida del software**

De las mejores prácticas, extraer modelos de cómo ejecutar esos procesos para evitar los problemas de la “crisis del software”

→ **CMM / CMMI - ISO/IEC TR 15504**

Definir estándares menores para dibujar criterios unificadores en requisitos, pruebas, gestión de la configuración, etc.

→ **IEEE 830 - IEEE 1362 - ISO/IEC 14764 ...**

1.- Introducción a la Ingeniería del Sw

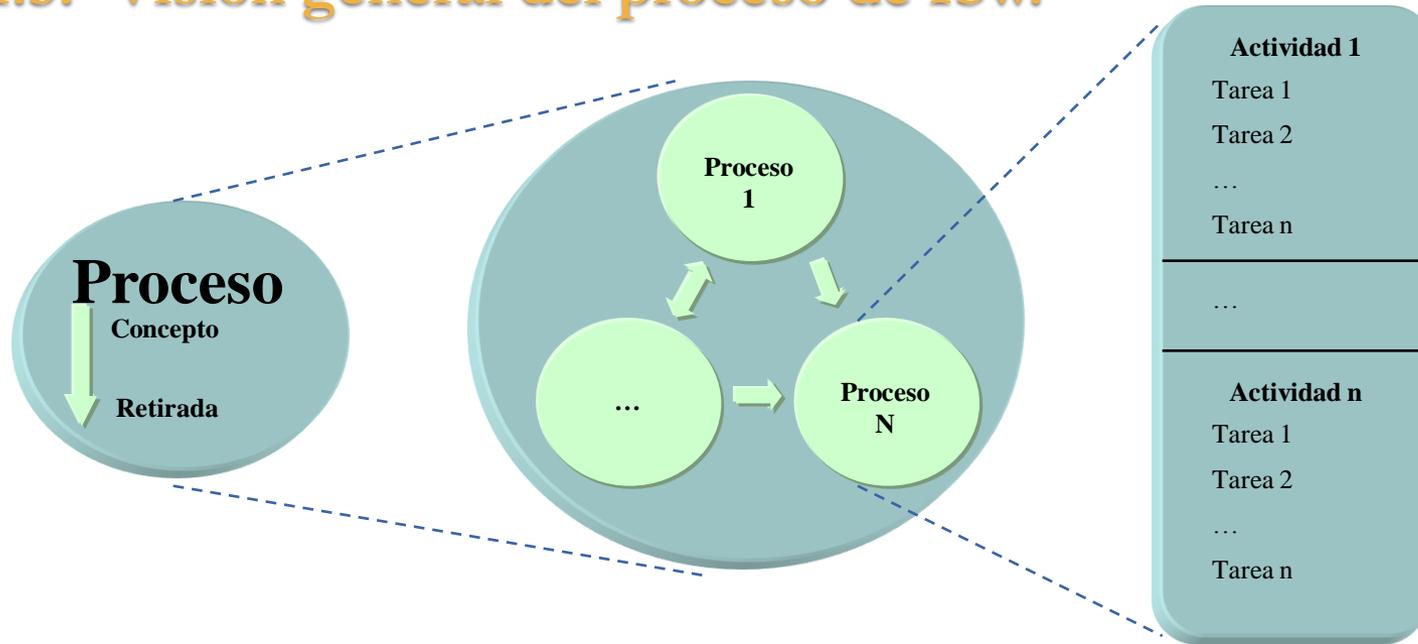
1.4.- La Ingeniería del Sw.

Más definiciones de ISw...

- ❑ “La ISw es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales” (*Fritz Bayer, 1968 (conferencia NATO)*).
- ❑ “Una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza” (*Sommerville 2002*).
- ❑ “Disciplina para producir software de calidad desarrollado sobre las agendas y costes previstos y satisfaciendo los requisitos”. (*S. Schach 1990, Software Engineering*)

1.- Introducción a la Ingeniería del Sw

1.5.- Visión general del proceso de ISw.



- Con independencia del área de aplicación, tamaño o complejidad del proyecto, cualquier sistema se encontrará al menos en una de las siguientes fases genéricas:

Definición ~ Análisis (del sistema, del sw.) ¿Qué debe hacer el sistema?

Desarrollo ~ Diseño, codificación, prueba ¿Cómo construir el sistema?

Mantenimiento.

**Metodología de Planificación,
Desarrollo y Mantenimiento de sistemas de información
MÉTRICA. Versión 3**

Planificación de Sistemas de Información (Proceso PSI)
Estudio de Viabilidad del Sistema (Proceso EVS)
Análisis del Sistema de Información (Proceso ASI)
Diseño del Sistema de Información (Proceso DSI)
Construcción del Sistema de Información (Proceso CSI)
Implantación y Aceptación del Sistema (Proceso IAS)
Mantenimiento del Sistema de Información (Proceso MSI)

2.- Sistemas de Información

2.1.- Concepto de Sistema.

□ “Un modelo formado por una serie de elementos interrelacionados entre sí, que opera en un entorno cambiante y con unos determinados objetivos”.

□ Elementos de un sistema:

Los **componentes** del sistema.

Las **relaciones** entre ellos, que determinan la estructura del sistema.

El **objetivo** del sistema. El **entorno** del sistema: aquello que lo rodea, dentro del cual está ubicado.

Los **límites** del sistema: la frontera entre lo que es el sistema y lo que constituye el entorno.

2.- Sistemas de Información

2.2.- Sistema de Información.

Una definición de Sistema de Información:

“Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) para desempeñar su actividad de acuerdo a su estrategia de negocio.” (Andreu et al. 91)

Otras definiciones de SI enfatizan que el objetivo es proporcionar información de calidad:

“El objetivo del SI es ayudar al desempeño de las actividades en todos los niveles de la organización, mediante el suministro de la información adecuada, con la calidad suficiente, a la persona apropiada, en el momento y lugar oportunos, y con el formato más útil para el receptor.”

2.- Sistemas de Información

2.3.- Sistema de Información Automatizado y Empresarial.

Un SI no necesita, para existir, estar obligatoriamente basado en el uso de ordenadores.
El SI existe siempre, esté mecanizado o no.

La aplicación del ordenador a los SI produce los sistemas de información basados en computadora o Sistemas de Información Automatizados (SIA).



“El **SI empresarial** es el sistema encargado de coordinar los flujos y registros de información necesarios para llevar a cabo las funciones de una empresa de acuerdo con *su planteamiento* o **estrategia de negocio**” (Andreu 96)



Los procedimientos y las prácticas habituales de trabajo.
La información.
Las personas o usuarios.
El **equipo de soporte** para la comunicación, el procesamiento y el almacenamiento de información.

2.- Sistemas de Información

2.4.- Seguridad y protección de la Información.

Ley:

[Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.](#)

https://www.agpd.es/portalweb/canaldocumentacion/legislacion/estatal/common/pdfs/Ley-15_99.pdf

Real Decreto:

[Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal.](#)

https://www.agpd.es/portalweb/canaldocumentacion/legislacion/estatal/common/pdfs/RD_1720_2007.pdf

Artículo 1. Objeto

La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.



3.- Herramientas CASE

Método

Procedimiento aplicado rutinariamente para alcanzar un objetivo.

El procedimiento define los resultados a alcanzar y el camino que conduce a ellos.

Herramienta

Producto software que libera al ingeniero de software de acciones que generan los resultados definidos por los métodos.

CASE (Computed Aided Software Engineering): Conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida.

- ❑ Gestión del proyecto
(*planificación, estimación y control*)
- ❑ Desarrollo del software
(*análisis, diseño, implementación, validación*)
- ❑ Mantenimiento del software.

AUTOMATIZACIÓN DEL DESARROLLO DE SW.:

Productividad del equipo ↑↑

Calidad del Software ↑↑

- ❑ INCREMENTAR
Reusabilidad del software.
- ❑ REDUCIR
Costes de desarrollo y mantenimiento.
- ❑ AUTOMATIZAR/SIMPLIFICAR
Gestión del proyecto.
Desarrollo del sw. (permitir aplicación met. estructuradas; prototipos; desarrollo “visual”)
Mantenimiento del software (Incluyendo la automatización y estandarización de la documentación y de su mantenimiento)

3.- Herramientas CASE

Características de una Herramienta CASE

Soporte gráfico para varias técnicas (DFD, E/R, STD, modelos OO, etc.)

Control de errores “Consistencia”: Unicidad identificadores, reglas metodología, etc.

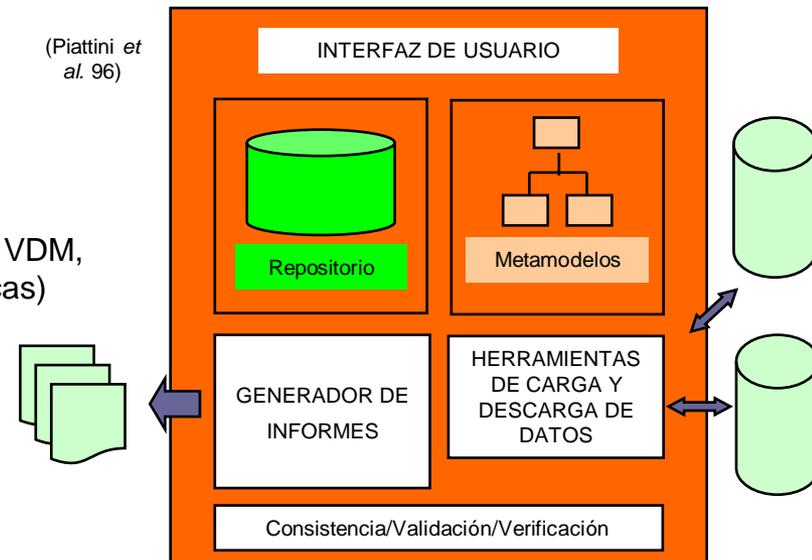
Validación entre diferentes modelos:

En una fase (p.e. entre DFD y E/R)

En varias fases (DFD análisis y diseño; DFD diseño y Diagrama de estructuras).

- Soporte multiusuario.
 - Personalización.
 - Control de documentos y versiones.
 - Gestión de proyectos
 - Estadísticas de productividad y métricas del software.
 - Pruebas.
 - Simulación y prototipado.
 - Demostración correcciones especificaciones y/o software.
 - Generación de código.
- } Uso de técnicas formales (Z, VDM, Especificaciones algebraicas)

(Piattini *et al.* 96)



3.- Herramientas CASE

Taxonomía de Herramientas CASE

Clasificación por cobertura:

Toolkits

- Conjunto de herramientas software (integradas) que automatizan alguna fase del ciclo de vida (análisis, diseño, gestión, ...). Comparten la BD de soporte y la interfaz de usuario.

Workbenchs

- Conjunto de herramientas software integradas que automatizan el ciclo de vida en más de una fase, típicamente en todas las *fases de desarrollo*: análisis + diseño + implementación (incluyendo la documentación asociada). Es decir, asisten en todo el proceso y proporcionan un producto documentado y operativo. Además de compartir la BD de soporte y la interfaz de usuario, están basadas en una misma metodología.

Según grado de integración:

Toolkits (Juego de herramientas)

- Conjunto de herramientas software que automatizan alguna fase del ciclo de vida (análisis, diseño, gestión, ...). Comparten la BD de soporte y la interfaz de usuario. Integración baja.

Workbenchs (Banco de trabajo)

- Conjunto de herramientas software que automatizan el ciclo de vida en más de una fase, típicamente en análisis + diseño + implementación (incluyendo documentación asociada). Asisten en todo el proceso y proporcionan un producto documentado y operativo. Además de compartir la BD de soporte y la interfaz de usuario, están basadas en una misma metodología. Integración media.

IPSE (*Integrated Project Support Environment*)

- Integración alta. Cubren todo el ciclo de vida, la *gestión de proyectos y de la configuración*.

3.- Herramientas CASE

Taxonomía de Herramientas CASE

Según las fases del ciclo de vida que abordan:

Upper CASE: Herramientas de apoyo a las **primeras fases**

- Métricas del software.
- Estimación de costes.
- Planificación temporal.

Medium CASE: Herramientas de apoyo a las **fases centrales.**

- Análisis.
- Diseño.

Lower CASE: Herramientas de apoyo a las **últimas fases.**

- Implementación (generación de código).
- Pruebas (caja blanca y caja negra).
- Mantenimiento.
- Documentación de la implementación y documentación para el usuario final.

ICASE (*Integrated-CASE*)

Contienen elementos de *Upper* y *Lower* CASE: Contemplan todo el ciclo de desarrollo.

Reverse Engineering (Ingeniería inversa) - Tema 9.- Mantenimiento del Software.

CARE (*Computer Aided Requirements Engineering*)

3.- Herramientas CASE

Taxonomía de Herramientas CASE

Según su funcionalidad:

*Herramientas de **gestión de proyectos*** ayudan a la planificación y seguimiento del proyecto. Por Ejemplo SuperProject

Planificación: agenda de desarrollo.

Estimación: costes, duración, esfuerzo.

Control: productividad, calidad.

*Herramientas de **análisis y diseño***. System Architect

*Herramientas de **prototipado y simulación***.

*Herramientas de **programación***.

Editores dirigidos por la sintaxis (cabeceras de subrutinas, palabras clave, indentación, nomenclatura de variables, ...)

Generadores de estructuras de programas.

Entornos integrados de desarrollo para soporte de un lenguaje (editor, compilador, depurador).

*Herramientas de **integración y pruebas***.

Analizadores estáticos.

Depuradores.

Generadores de datos.

Comparadores (e.g. de ficheros).

*Herramientas de **soporte***.

*Herramientas de **mantenimiento***.

Ingeniería inversa.

✓ Reingeniería.